Basi di Dati

Esercitazione JDBC

Ing. Paolo Cappellari

Sorgente di dati ODBC

Andare su: ***Avvio → Pannello di Controllo → Strumenti di Amministrazione → Origine dati (ODBC)***.

Sorgente di dati ODBC

DSN

a. utente: disponibile solo per l'utente che lo crea.
b. DSN sistema: disponibile per tutti gli utenti.

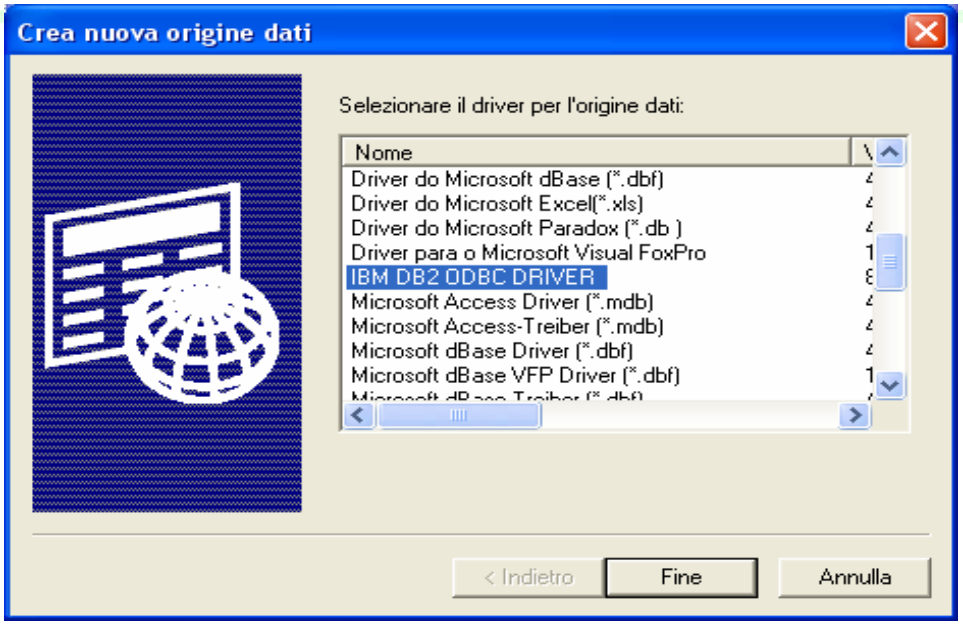
Click su ***Aggiungi***

Scelta del driver: "***IBM DB2 ODBC DRIVER***"

Scegliere il nome della sorgente: "***EsercSQL***"

Scegliere un (alias di) database, già esistente, cui connettersi: "***EsJDBC***"

Ok!



Esercizio 1 e 2

Considerando il seguente schema:

Fornitori (CodiceFornitore, Nome, Indirizzo, Citta)

Prodotti (CodiceProdotto, Tipo, Marca, Modello)

Catalogo (CodiceFornitore, CodiceProdotto, Costo)

1. Scrivere una applicazione Java che crea la tabella Fornitori.
2. Scrivere una applicazione Java che inserisce i seguenti Fornitori:

CodiceFornitore	Nome	Indirizzo	Citta
001	Ladroni	Via Ostense	Roma
002	Risparmietti	Viale Marconi	Roma
010	Teloporto	Via Roma	Milano

Connessione al database

```
import java.sql.*;
public class Esercizio_1_1 {
  public static void main(String[] arg){

    Connection con = null;
    try { // Caricamento del driver
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    }
    catch (ClassNotFoundException exClass) {
      System.err.println("Fallita connessione al database. Driver " +
        " non trovato");
    }

    String url = "jdbc:odbc:EsercSQL";
    String username = "";
    String pwd = "";
    try { // Apertura della connessione
      con = DriverManager.getConnection(url, username, pwd);
    }
    catch (SQLException exSQL) {
      System.err.println("Fallita connessione al database. La " +
        "sorgente dati (ODBC) non esiste.");
    }
```

Esercizio 1

1. Scrivere una applicazione Java che crea la tabella Fornitori.

```
import java.sql.*;
public class Esercizio_1_1 {
  public static void main(String[] arg){
…
  connessione al database
…
    try { // Esecuzione dell'interrogazione SQL
      Statement createTable = con.createStatement();
      createTable.executeUpdate( "CREATE TABLE Fornitori ("  +
                    "CodiceFornitore   VARCHAR (20)  NOT NULL, "
    +
                    "Nome              VARCHAR (20) NOT NULL, "  +
                    "Indirizzo         VARCHAR (30)         , "  +
                    "Citta             VARCHAR (20)         , "  +
                    "PRIMARY KEY( CodiceFornitore )          )");

      System.out.println("La tabella Fornitori e' stata creata.");
      createTable.close();
      con.close();
    }
    catch (SQLException exQuery){
      System.err.println("Errore nell'interrogazione.");
    }
```

Esercizio 2

2. Scrivere una applicazione Java che inserisce i fornitori

```
import java.sql.*;

public class Esercizio_1_2 {

// Array contenente i record da inserire
static String[] SQLData = {
      "('001', 'Ladroni',      'Via Ostense',   'Roma')",
      "('002', 'Risparmietti', 'Viale Marconi', 'Roma')",
      "('010', 'Teloporto',    'Via Roma',      'Milano')"
};

    public static void main(String[] arg){

…
   connessione al database
…
```

Esercizio 2

2. Scrivere una applicazione Java che inserisce i fornitori

```
…
try { // Esecuzione dell'interrogazione SQL
  Statement stmt = con.createStatement();
  int iRowCount = 0;
  for (int i = 0; i < SQLData.length; i++) {
    iRowCount += stmt.executeUpdate(
      "INSERT INTO Fornitori VALUES " + SQLData[i] );
  }
  System.out.println( iRowCount +
    " righe inserite nalla tabella Fornitori.");

  stmt.close();
  con.close();
}
catch (SQLException exQuery){
  System.err.println("Errore nell'interrogazione.");
}
…
```

Esercizio 3

3. Scrivere una applicazione Java che stampa, per ogni fornitore, l'elenco dei prodotti forniti (Marca, Tipo prodotto, Modello, Costo).

OUTPUT

```
Nome Indirizzo Citta
    Marca, Tipo, Modello, Costo
    ...
    Marca, Tipo, Modello, Costo
Nome Indirizzo Citta
    Marca, Tipo, Modello, Costo
    ...
    Marca, Tipo, Modello, Costo
...
```

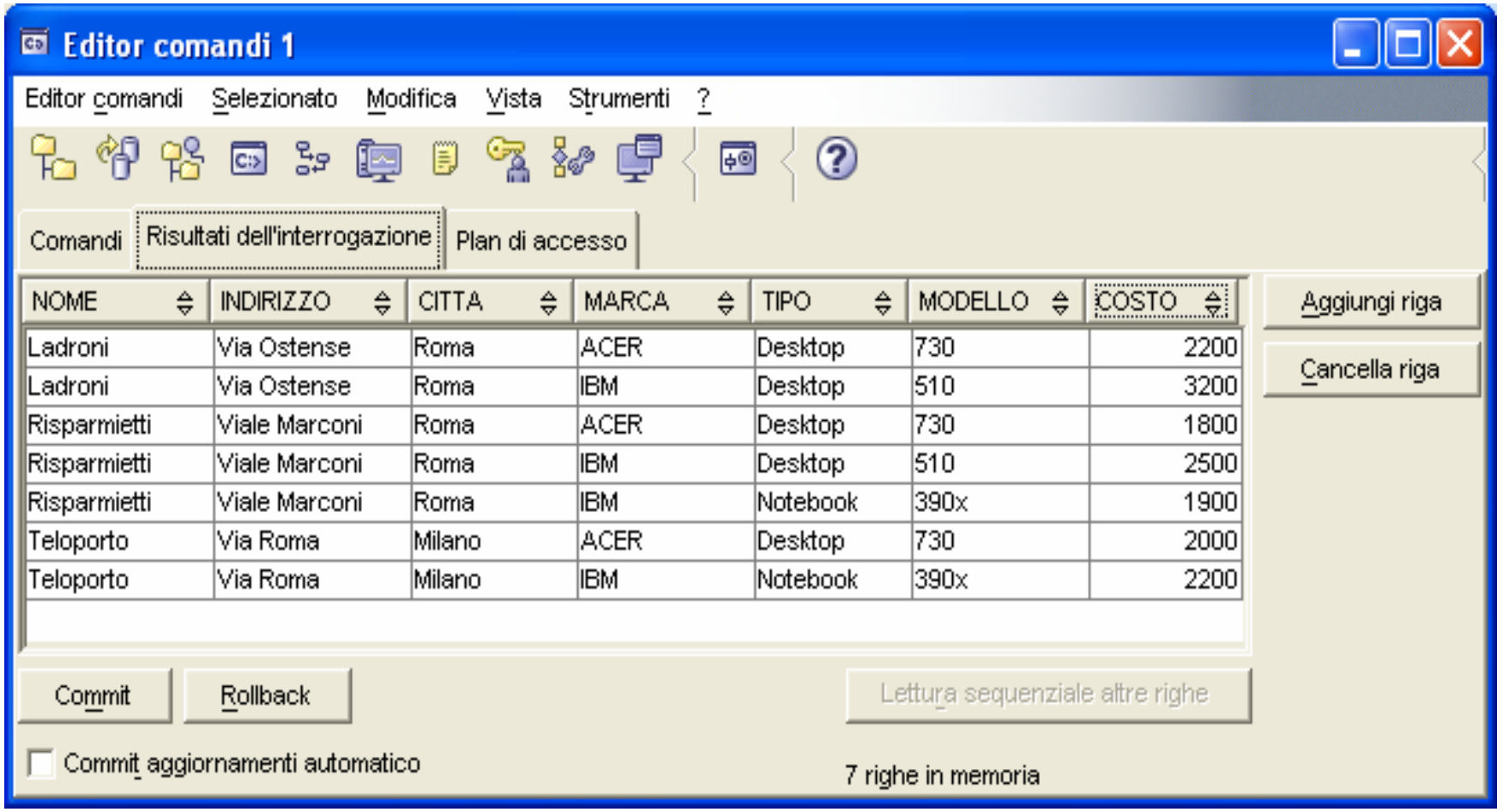
Esercizio 3

3. Scrivere una applicazione Java che stampa, per ogni fornitore, l'elenco dei prodotti forniti (Marca, Tipo, Modello, Costo).

```
import java.sql.*;
public class Esercizio_1_3 {
public static void main(String[] arg){
…
   connessione al database
…
   try { // Esecuzione dell'interrogazione SQL
     Statement query = con.createStatement();
     String queryString =
       "SELECT Nome, Indirizzo, Citta,  " +
       "  Marca, Tipo, Modello, Costo " +
       "FROM Fornitori AS F, Catalogo AS C, Prodotti AS P " +
       "WHERE C.CodiceFornitore = F.CodiceFornitore    " +
       "  AND C.CodiceProdotto = P.CodiceProdotto     " +
       "ORDER BY Nome, Marca, Tipo, Modello ";

     ResultSet result = query.executeQuery(queryString);
…
```

Esercizio 3


Editor comandi 1
Editor comandi
Selezionato
Modifica
Vista
Strumenti
?
Comandi
Risultati dell'interrogazione
Plan di accesso
NOME
INDIRIZZO
CITTA
MARCA
TIPO
MODELLO
COSTO
Ladroni Via Ostense Roma ACER Desktop 730 2200
Ladroni Via Ostense Roma IBM Desktop 510 3200
Risparmietti Viale Marconi Roma ACER Desktop 730 1800
Risparmietti Viale Marconi Roma IBM Desktop 510 2500
Risparmietti Viale Marconi Roma IBM Notebook 390x 1900
Teloporto Via Roma Milano ACER Desktop 730 2000
Teloporto Via Roma Milano IBM Notebook 390x 2200
Aggiungi riga
Cancella riga
Commit
Rollback
Lettura sequenziale altre righe
Commit aggiornamenti automatico
7 righe in memoria

Esercizio 3

```
…
// Elaborazione del risultato
  String nomeF, indirizzoF, cittaF, marcaP, tipoP, modelloP;
  int costoC ; String nomeF_old = "";
  while (result.next()){
    nomeF = result.getString("Nome");
    indirizzoF = result.getString("Indirizzo");
    cittaF = result.getString("Citta");
    if (!nomeF_old.equals(nomeF))
      System.out.println(nomeF + " " + indirizzoF + " " + cittaF );
    marcaP = result.getString("Marca");
    tipoP = result.getString("Tipo") ;
    modelloP = result.getString("Modello") ;
    costoC = result.getInt("Costo") ;  // sia costo un intero
    System.out.println("\t" + marcaP + " " + tipoP + "  " +
                               modelloP + " " + costoC);
    nomeF_old = nomeF ;
  } // chiudo while
  …
```

Esercizio 3

```
...
    result.close();
    query.close();
    con.close();
    } // chiudo try
    catch (SQLException exQuery){
      System.err.println("Errore nell'interrogazione.");
      System.err.println(exQuery.getErrorCode() + " " +
      exQuery.getSQLState() + "\n" + exQuery.getMessage() );
    }
  } // chiudo main
} // chiudo classe
```

Basi di Dati

Esercitazione ER

Ing. Paolo Cappellari

Esercizio 1

Definire uno schema Entity-Relationship che descriva i dati di una applicazione relativa alla programmazione cinematografica nei cinema. Nei cinema vengono proiettati film che hanno un regista e degli attori.

Sono di interesse:

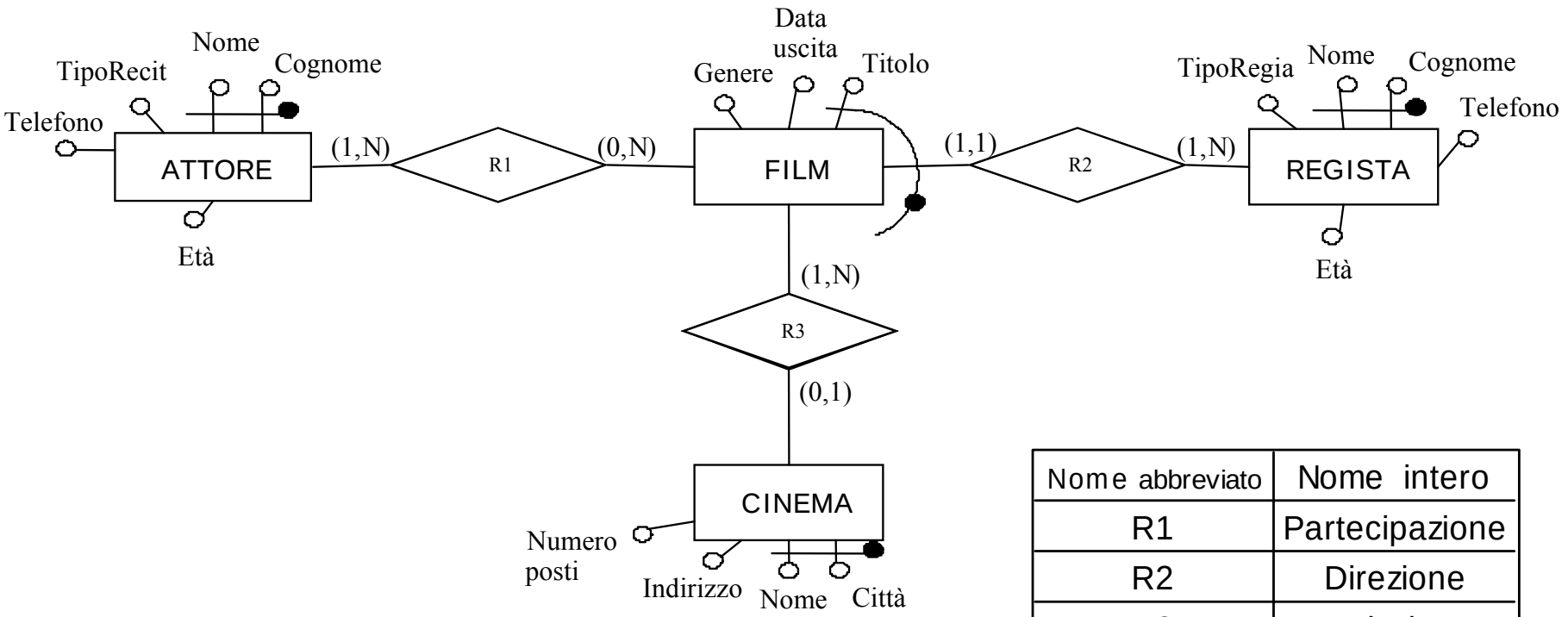
- per i film: il titolo, il genere, il regista, la durata, la data di uscita
- per i cinema: il nome, la città, l'indirizzo, il numero di posti
- per gli attori: il nome, il cognome, l'età, il telefono, il tipo di recitazione (comico, drammatico,…)
- per i registi: il nome, il cognome, l'età, il telefono, il tipo di regia (comico, drammatico,…)

Inoltre:

- ogni film ha uno e un solo regista
- ogni film ha zero o più attori
- ogni film è programmato presso uno o più cinema

Indicare le cardinalità delle relazioni e un identificatore per ciascuna entità.

Esercizio 1



Nome abbreviato	Nome intero
R1	Partecipazione
R2	Direzione
R3	Proiezione

Esercizio 2

Definire uno schema Entity-Relationship che descriva i dati di una applicazione relativa ai listini prezzi di un insieme di case automobilistiche. Sono di interesse:

- Le case produttrici, con nome (identificante) e indirizzo.
- I modelli (ad esempio la Punto o la Golf), con nome, anno di lancio e segmento di mercato (codificato con una lettera e con una breve descrizione: ad esempio, al segmento “A” corrisponde la descrizione “utilitaria”). Il nome identifica univocamente insieme alla casa produttrice.
- Le versioni dei modelli , identificate attraverso il nome della casa, quello del modello e un nome specifico (ad esempio la Fiat Punto 75S). Per ogni versione sono rilevanti il prezzo, il motore, la cilindrata, la potenza, il numero di porte e la velocità massima. Ogni versione di modello ha uno ed un solo motore.
- I motori (ad esempio il motore Fire 1000), identificati attraverso un codice e con le seguenti proprietà: cilindrata, numero cilindri e potenza. Possono esistere motori (attualmente) non utilizzati in alcun modello.